



**Manchester
Metropolitan
University**

Erfani, T, Mokhtar, H and Erfani, R (2017) Self-adaptive agent modelling of wind farm for energy capture optimisation. *Energy Systems*, 9 (1). pp. 209-222. ISSN 1868-3967

Downloaded from: <https://e-space.mmu.ac.uk/623653/>

Version: Accepted Version

Publisher: Springer

DOI: <https://doi.org/10.1007/s12667-017-0243-y>

Please cite the published version

<https://e-space.mmu.ac.uk>

Self-Adaptive Agent Modelling of Wind Farm for Energy Capture Optimisation

Tohid Erfani^a, Hamid Mokhtar^c, Rasool Erfani^b

^aUniversity College London, Department of Civil, Environmental and Geomatic Engineering, London, UK

^bManchester Metropolitan University, Department of Mechanical Engineering, Manchester, UK

^cMechanical and Aerospace Eng., Faculty of Engineering, Monash University, Clayton Vic 3800, Australia

Abstract

Typical approaches to wind turbines placement problem take into account the wind distribution and wake effects to maximise the total aggregate farm's energy production in a centralised *top-down* optimisation problem. An alternative approach, however, is yet to be addressed as the problem can be instead modelled in a decentralised *bottom-up* manner emulating a system of self-adaptive agents. The potential advantages of this is that it offers easier scalability for high dimension problems as well as it enables an easier adaptation to the complex structure of the design problem. This paper contributes to this and presents an evolutionary algorithm to model and solve the wind farm layout design problem as a system of interrelated agents. The framework is applied to problems with different complexities where the quality of the results is examined. The convergence and scalability of the suggested technique indicate promising results for small to large scale wind farms, which, in turn, encourage the application of such an evolutionary based algorithm for real world wind farm design problem.

Keywords: Wind farm layout design; Agent based modelling; Evolutionary algorithm; Self adaptive agents.

1. Introduction and motivation

The key elements in maximising the energy output of a wind farm are wind variation and turbine interaction. The placement of each wind turbine may adversely affect the performance of other turbines in the wind farm due to wake formation. The wind farm layout problem is concerned with the design and development of an efficient wind turbine placement on a farm site considering wind variations, design objectives, technical constraints and wake effects (Herbert-Acero et al., 2014; González et al., 2014). This design problem is formulated as a constrained optimisation problem where efficiency is defined as to maximise the total wind farm energy output affected by the wind, land and cost constraints. A comprehensive review

Email addresses: t.erfani@ucl.ac.uk (Tohid Erfani), hamid.mokhtar@monash.edu (Hamid Mokhtar), r.erfani@mmu.ac.uk (Rasool Erfani)

on these approaches reveals a common perspective in defining the optimisation problem (Chehouri et al., 2015; Khan & Rehman, 2013). In almost all of the proposed approaches (one may refer to the modelling formulations presented by Kusiak & Song (2010); Pookpunt & Ongsakul (2013); Chowdhury et al. (2012); Feng & Shen (2015)), the wind farm energy capture is formulated based on aggregate function of farm's power production and single or multiobjective optimisation search algorithms are exploited to obtain the wind turbines' optimal locations. This type of modelling, as a result of a central ordering influence over its components, is called a centralised *top-down* approach. Decentralised based modelling, as an alternative, introduces a *bottom-up* perspective to system modelling avoiding any commanding influence over its components. The components are self-adaptive agents which use local information to accomplish global goals. Following the agent-based problem terminologies, in the top-down approaches, the 'best' solution is defined as the optimal solution to a single aggregated optimisation problem as opposed to bottom-up methods that 'best' is defined as the optimal solutions of independent but connected optimisation problems. Further, in bottom-up approaches, agents have partial information about the whole model and collectively solve a common task. The potential advantages of using bottom-up techniques are trifold: (1) introducing flexibility in modelling, (2) enabling easier adaptation to the complex requirements and changing environment, and (3) offering reduced processing time, robustness, and scalability.

These capabilities have been demonstrated in various applications such as software engineering, traffic analysis, telecommunications network and resource allocation and management (e.g. look at Forrest (1990); Erfani & Erfani (2015b,a); Ding et al. (2016)). In literature, however, decentralised modelling has not been employed for modelling the wind farm for maximising its energy output. This paper applies self-adaptive agent-based technique to wind farm energy production optimisation problem. In this paper, self-adaptive agents represent the turbines and they collectively solve a common task of maximising energy production by communicating with others. An evolutionary algorithm is developed considering this dynamics. For this, we first explain the wind farm components briefly and build up the optimisation model in Section 2. Development of the algorithm and the strategies come in Section 3. In Section 4 the proposed algorithm is applied to numerical problems with different complexities where the results are compared with those found in literature. Section 4 also investigates the applicability of the algorithm on a real problem in Manjil wind farm plant in north Iran. In Section 5 the scalability of the proposed algorithm is analysed and its performance is compared with that of the centralised aggregated optimisation algorithm. Section 6 concludes the paper with extra discussion for future work.

2. Wind farm modelling

While there are various design factors in wind farm layout problem (such as turbine types and soil load-bearing capacity (Rahbari et al., 2014)), the most important elements are the wind characteristics and the

speed loss behind each turbine due to the wake formation (Kusiak & Song, 2010). This section incorporates these elements into the farm modelling and builds up the mathematical models along with the optimisation formulation.

2.1. Wake model

In this study, due to its trade-off between simplicity and accuracy, we follow the Jensen wake model (Jensen, 1983; Mosetti et al., 1994). Let n be the number of wind turbines in a given wind farm with i th wind turbine location coordinates $\phi_i = (x_i, y_i)$. The wind speed experienced by downstream turbine i is reduced as a result of a wake of the upstream wind turbine j and is given by

$$v_{ij} = v_0(1 - \delta v_{ij}), \quad \text{with,} \quad \delta v_{ij} = \frac{2\alpha_j}{\left(1 + \frac{\kappa_j d_{ij}}{R}\right)^2}, \quad (1)$$

where v_0 is the mean wind speed, κ_j the turbine wake spreading constant, and R and d_{ij} respectively define the rotor radius and wake downstream distance between turbine i and j . $\alpha_j = 1/2(1 - \sqrt{1 - CT_j})$ is turbine's induction factor which depends on turbine thrust coefficient CT_j . Let \mathcal{N}_i be the set of upstream wind turbines respect to turbine i in wind direction θ . When a turbine is affected by the wakes of multiple turbines,

$$v_i = v_0 \left[1 - \sqrt{\sum_{j \in \mathcal{N}_i} \left(1 - \frac{v_{ij}}{v_0}\right)^2} \right], \quad (2)$$

is the wind speed v_i experienced by turbine i downstream of turbines \mathcal{N}_i (Katic et al., 1986).

2.2. Wind model

Two important factors in wind variation modelling for farm layout design are wind speed v and direction θ . These affect the wake production behind the turbines and therefore an optimal turbine placement within the farm. Wind speed v is measured as m/s in direction $0 < \theta \leq 360$, where (from) north (N) and east (E) blowing winds are defined as 90° and 0° , respectively. Given the wind data (speed and direction), following Zhang et al. (2013), multivariate and multimodal wind distribution (MMWD) model is used in this paper to more realistically approximate the wind variability as a joint probability distribution of the wind speed and wind direction, $p_w(v, \theta)$. Reader is referred to Zhang et al. (2013) for additional details.

2.3. Optimisation model

Given the wind variation model, $p_w(v, \theta)$, and power generation model $P_i(v)$ for turbine i , the wind farm expected power production \hat{EP} is given by,

$$\hat{EP} = \sum_{i=1}^n \int_{\theta} \int_v P_i(v) p_w(v, \theta) dv d\theta \quad (3)$$

where summation over i is added to account for the total number of wind turbines n . $P_i(v)$, power generated by turbine i , is generally proportional to cubic of the wind speed (e.g. look at Manwell et al. (2010)) and wind model $p_w(v, \theta)$ is discretised by s wind scenarios. \hat{EP} is, therefore, approximated by,

$$\begin{aligned}\mathbf{EP} &= \sum_{i=1}^n EP_i, \\ \mathbf{EP} &= \sum_{i=1}^n \sum_{j=1}^s P_i(v_j) p_w^j(v_j, \theta_j).\end{aligned}\tag{4}$$

As a result of the wake effect due to turbine locations respect to one another, the wind speed is reduced (look at Equation 2). As a consequence, \mathbf{EP} value (as a function of wind speed) varies according to the turbines layout design. $\phi_i = (x_i, y_i)$, the coordinate of turbine i , is a continuous variable and its value is subject to the upper and lower bound on the area of the farm. For a rectangular farm with length l and width h , $0 \leq x_i \leq l$ and $0 \leq y_i \leq h$ are realistic bound constraints for turbine locations within the wind farm. The optimisation problem is to find the best turbines locations in order to maximise the total power production as formulated by,

$$\begin{aligned}& \max \quad \mathbf{EP} \\ & \text{s.t.} \\ & 0 \leq x_i \leq l \quad \forall i = 1, \dots, n \\ & 0 \leq y_i \leq h \quad \forall i = 1, \dots, n \\ & (x_i - x_j)^2 + (y_i - y_j)^2 \geq fR^2 \quad \forall i, j = 1, \dots, n, i \neq j\end{aligned}\tag{5}$$

where f is a safety factor that is usually set by turbine's manufacturer using the rotor radius R to ensure that the adjacent turbines are sufficiently far from each other for interaction reduction e.g., in wind turbulences, thus diminishing the hazardous loads on the turbine (Kusiak & Song, 2010).

with f as the safety factor. Inline with the work of Mosetti et al. (1994), in this study, it is assumed that the cost of the wind farm only depends on the number of wind turbines. Since the number of wind turbines is given and it is fixed, the cost function is omitted from modelling. This fits for our purpose of demonstrating the effectiveness of the proposed algorithm.

3. Wind farm optimisation using self-adaptive agents

Let EP_i be the expected power production of turbine i defined in Equation 4, and define problem T_i , $i = 1, \dots, n$ for each turbine i as,

$$\underset{\phi_i}{\text{maximise}} \quad EP_i(\phi), \tag{T_i}$$

with $\phi = (\phi_1, \dots, \phi_n) \in \mathbb{R}_+^n$. Each turbine i controls vector $\phi_i \in \mathbb{R}^{n_i}$ to optimise the objective function EP_i . Let \mathbf{T} be the problem formed by all T_i s. We find ϕ by simultaneously solving n optimisation problems T_i , $i = 1, \dots, n$.

Algorithm 1: Search algorithm for wind turbine placement

```

1  Set MaxGen convergence tolerance,  $\epsilon$ , number of turbines (number of population),  $n$ , and population
   size,  $m$ ;
2  For each turbine  $i$ ,  $[o_i]_{m \times 1} \leftarrow$  Randomly initialise  $m$  coordinates;
3   $\forall i = 1, \dots, n$ , Let  $pop_i = (o_i, o_{-i})$ ;
4  while  $\eta > \epsilon$  and Not MaxGen do
5      for turbine  $i = 1$  to  $n$  do
6          for each individual  $\phi_i^k$  in  $pop_i$  and  $k=1$  to  $m$  do
7              Take wind speed,  $v_0$ , and wind direction,  $\theta$ ;
8              Set  $\mathcal{N}_k$  an empty set;
9              for turbine  $j \neq i$  do
10                 if Turbine  $j$  is upstream of  $k$  then
11                      $\mathcal{N}_k \leftarrow \mathcal{N}_k \cup \{j\}$ ;
12                 Calculate  $d_{kj}$  and  $\delta v_{kj}$ ;
13             Calculate velocity ( $v_i^k$ ) and expected power production ( $EP_i^k$ ) associated with  $k$ th row;
14              $[pop_i]_k = [pop_i]_k \cup \phi_j^k, j \in \mathcal{N}$ ;
15             Randomly pick  $p_{s1} \neq p_{s2} \neq p_{s3} \neq p_k$  from  $pop_i$ ;
16              $p_b \leftarrow$  reproduction ( $p_{s1}, p_{s2}, p_{s3}$ );
17              $\phi_i^* \leftarrow$  The best individual coordinate in  $pop_i$ ;
18          $\forall i = 1, \dots, n$ ,  $pop_i = (o_i, \bar{o}_{-i}^*)$ ;
19     Update  $\eta$ ;
```

Let ϕ_{-i} be a vector containing the decision variables of all turbines involved in problem T_i excluding that of the turbine i . The search Algorithm 1 is described by n different search trajectories performing simultaneously as follows. Since each turbine i is associated with one problem T_i , the algorithm requires n number of populations. $[o_i]_{m \times 1}$ is a vector populated by turbine i 's m different locations $([\phi_i^1, \dots, \phi_i^m]^t)$ and $[o_{-i}]_{m \times n-1}$ is a coordinate matrix of other turbines excluding that of turbine i . ϕ_i^k is the k th row in pop_i for turbine i . Given wind direction θ , for each individuals k (ϕ_i^k) in pop_i , a neighbourhood set \mathcal{N}_k is formed. \mathcal{N}_k defines the set of all upstream turbines of turbine k for wake calculation. For the neighbouring turbines, the distance d and the retarded speed is computed. In Line 14, pop_i is updated with regards

to individual k 's neighbours ($[pop_i]_k$ is the k th row of pop_i). All individuals in each population i undergo reproduction in each generation g of searches (Line 15-16) using three mutually exclusive individuals from the population. The reproduction in Line 16 is differential evolution following the work of Price et al. (2005) and is explained in next Section. In case of violation, constraint violation techniques are used to treat them (Line 16). pop_i is sorted in decreasing order based on the objective value (maximisation), and ϕ_i^* , the best individual in pop_i is stored in the first row of pop_i (Line 17). At the end of each generation g , all turbines share their best individuals (coordinate) to form the updated population for next generation $g + 1$. The position of each turbine is updated in each generation using its last position in the farm. Turbine's position changes in each generation if, compared to its former position, new position contributes more to the power production function. In Line 18, $[o_{-i}^*] := \mathbf{1}_{m \times 1} \times [\phi_1^*, \dots, \phi_{i-1}^*, \phi_{i+1}^*, \dots, \phi_n^*]$ is a $m \times n - 1$ identical rows matrix. The rows in o_{-i}^* contain the ϕ_i^* that migrates to the population of the others and remains fixed (shown by $\bar{\phi}_i^*$) for the next generation, and this continues through generation. An example of population sharing scheme is illustrated in Figure 1 for three turbines a , b and c . For example, in pop_a , ϕ_b^* and ϕ_c^* are the best individuals (location) for turbine b and c in previous generation and are fixed in pop_a shown by $\bar{\phi}_b^*$ and $\bar{\phi}_c^*$. Due to n different search trajectories, the algorithm allows independent search for turbines that, over successive generation, leads to an optimal solution.

4. Experimental setup, results and discussion

We have tested the proposed algorithm performance on three problems with different complexities. Wind direction θ is such that for example (from) north (N) and (from) east (E) blowing winds are defined as 90° and 0° , respectively. For convergence, the algorithm uses a variance-based stopping criterion η (Sinha et al., 2014) defined by,

$$\eta = \sum \frac{\sigma_{V_{ic}}^2}{\sigma_{V_{io}}^2},$$

where $\sigma_{V_{ic}}^2$ and $\sigma_{V_{io}}^2$ denotes the variances of agents i 's decision variable ϕ_i (coordinate of turbine i) in the current (c) and original (o) population, respectively. The value of η is restricted between 0 and 1 and upon convergence it is expected that this value converges to zero. Therefore, if η is less than ϵ then the algorithm terminates, otherwise, the algorithm continues until $MaxGen$ is reached.

For constraint handling of load safety factor, while other techniques can be easily applied, the constraint domination technique (Deb, 2000) is adopted. The procedure ranks the feasible individuals based on their objective values while the infeasible individuals are ranked only based on their extent of constraint violation.

For reproduction stage, as noted earlier, Differential Evolution (DE) (Price et al., 2005) is chosen as it provides promising results on several test problems (Neri & Tirronen, 2010). In DE, corresponding to each

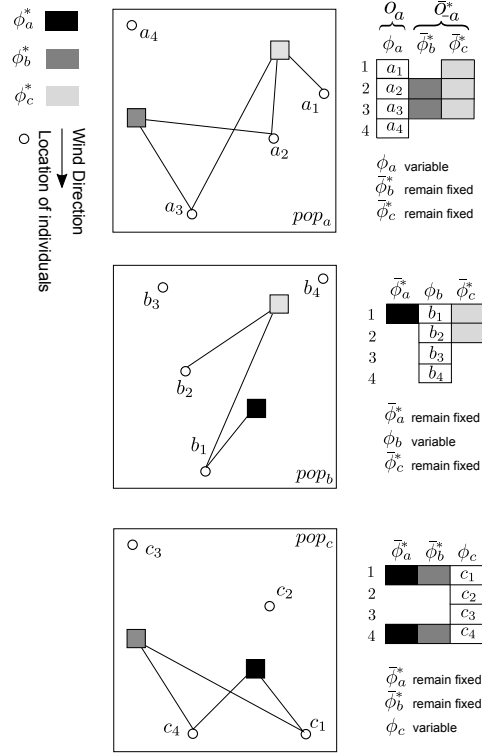


Figure 1: Exchanging the best individual values between populations at the end of each generation for three turbines a , b and c where $m = 4$. ϕ_a^* is fixed in first column and all rows of pop_b and pop_c , ϕ_b^* is fixed in second column and all rows of pop_a and pop_c , and finally ϕ_c^* is fixed in third column and all rows of pop_a and pop_b . The fixing is subject to each individual's neighbour \mathcal{N} . $\bar{\phi}^*$ shows the best individuals that migrates to the population of the others and remains fixed.

member of population p_k , three individuals p_{s1} , p_{s2} and p_{s3} are randomly chosen from o_i and new vector p_c is created by adding the weighted difference of p_{s2} and p_{s3} to the p_{s1} given by,

$$p_c = p_{s1} + MR(p_{s2} - p_{s3}),$$

where MR is the mutation rate. p_c is accepted as a new vector p_b if the following is satisfied

$$p_b = \begin{cases} p_c & \text{if } rand(0,1) \leq CR \\ p_k & \text{o.w.} \end{cases} \quad (6)$$

CR is the probability of crossover and $rand$ is pseudo random number between 0 and 1.

The population size for each agent is $m = 5$, the maximum number of generation is $MaxGen = 200$ and following Price et al. (2005), mutation and crossover rates within DE are $MR = 0.7$ and $CR = 0.5$, respectively. The convergence is assumed with tolerance of less than $\epsilon = 10^{-3}$. Wind farm efficiency ζ is calculated for further analysis as the ratio between the optimal **EP** and maximum power production in an ideal scenario that the wake interactions among turbines are neglected. To account for variability of each run and initial locations of each turbine, we randomly initialised each population and solved each problem 30 times. The coordination, wind farm energy production and the convergence rate results are, therefore, averaged over 30 runs to show the average location of each turbine, the average of wind farm energy production and the average of the algorithm convergence rate.

4.1. Numerical problems - cases 1 and 2

The problems in this section are introduced in the literature by Mosetti et al. (1994). The farm site is stretched 2000m to north and 2000m to east. In the first case, there are 30 wind turbines, the wind is unidirectional coming from north ($\theta = 90^\circ$) and has the intensity of $v = 12$ m/s. In the second case, there are 39 turbines, the wind is coming from 36 equally ranged direction from 0° to 360° with equally probable occurrence and the wind speed is constant at $v = 12$ m/s. In both cases, the rotor radius $R = 20m$, thrust coefficient $CT = 0.88$ and wake spreading constant $\kappa = 0.1$. As a loading safety factor constraint, it is required that each turbine is separated from the others by at least 5 times the rotor diameter. In both cases, the power generation for each identical turbine follows $EP_i = 0.3v_i^3$. We have used the studies of using genetic algorithm (GA) by Grady et al. (2005) and particle swarm optimisation (PSO) by Pookpant & Ongsakul (2013) to examine the quality of the proposed agent based method's results. Figure 2 shows the position of each turbine, their power production and the convergence behaviour. The convergence figures accompanied with Table 1 confirm a promising performance of the proposed agent based method.

4.2. Real world wind farm site layout problem - case 3

To further investigate the applicability of the proposed approach, we use real onshore wind farm layout problem in Manjil, Iran with a wind farm site area stretched 2500m to the east and 1200m to the north.

Table 1: Test cases 1 and 2: Farm's efficiency results of the proposed agent based approach as well as three other methods. The results are averaged over 30 runs.

Measures	Case 1	Case 2
Efficiency ζ		
GA (Grady et al., 2005)	92.01%	85.17%
PSO (Pookpant & Ongsakul, 2013)	92.01%	89.80%
Agent model (current study)	96.72%	89.47%
Output power (kW)		
Agent model (current study)	15043.2	18090.57
Ideal	15552	20217.6

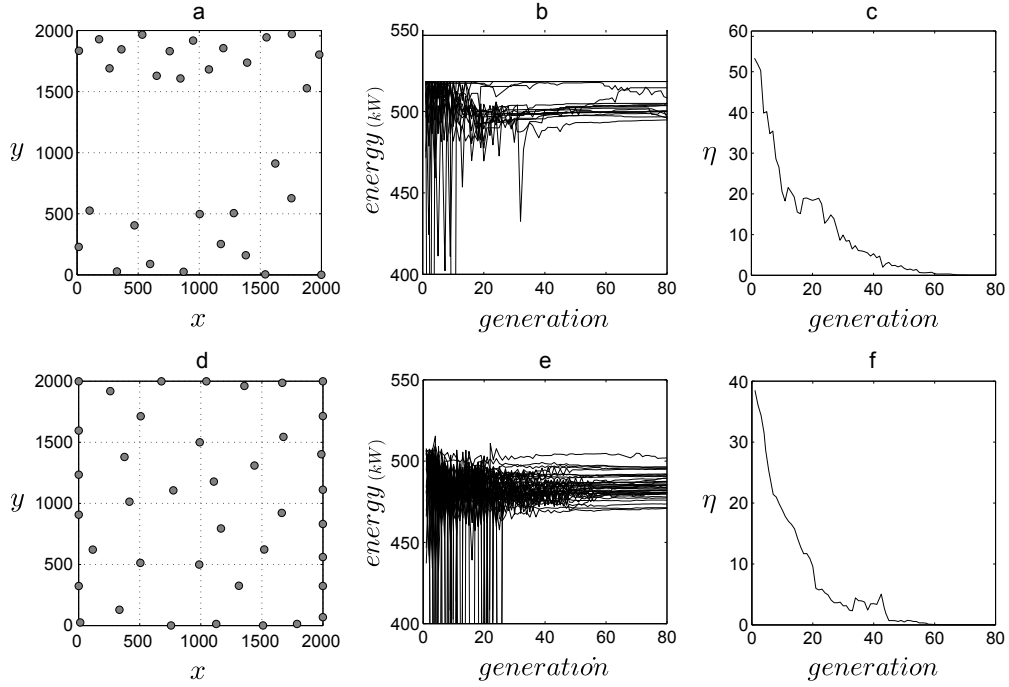


Figure 2: The positions, average power production and convergence behaviours of the numerical problems. The top row are the results for the first case (n=30 turbines), and bottom row shows the second case results (n=39 turbines).

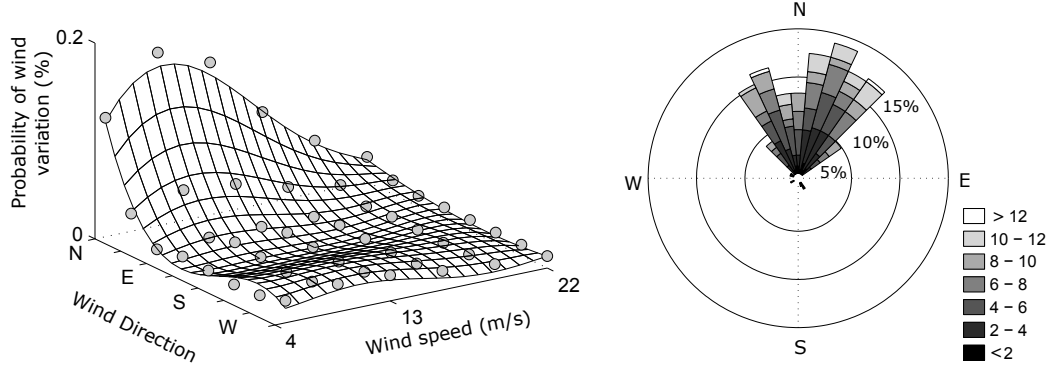


Figure 3: Manjil's wind variation model and its rose plot. *Direction* indicates the direction from which the wind blows. The gray dots are the sampled using LHS and used for this study.

The site has 57 identical $P_{rated}=500$ kW turbines with rotor radius $R = 19.5$ m, $CT = 0.88$ and $\kappa = 0.05$. Loading safety factor constraint requires that each turbine is separated from the others by at least five times the rotor diameter. The cut-in, rated and cut-out speed of each turbines used for energy production model are 4 m/s, 14 m/s and 25 m/s, respectively¹. Using 20 years (1993 - 2014) wind intensity and direction data from Iran Meteorological Organization² for Manjil station, we have exploited the MMWD approach proposed by Zhang et al. (2013) to model the wind variation. The variation is shown in Figure 3. The prevalent wind is blowing from north (N) and north-east (NE) direction with the average speed of 13 m/s. We have implemented Latin Hypercube Sampling (LHS) and have extracted $s = 46$ scenario sample points (shown as gray circle in Figure 3) to approximate $p_w(v, \theta)$ in Equation 3.

Having implemented the algorithm, Figure 4.a shows the optimal location of each turbine with its associated power output. The figure describes that due to the high occurrence probability of wind direction from NE, the higher power is generated from the wind turbine positioned in NE corner (more than 285 kW each). As the turbines are moved towards southern part of the farm, due to lower intense wind, their power production is reduced. Figure 4.c indicates that the convergence is achieved in less than 80 generations with the total wind farm power production of 15621.62 kW with farm efficiency of 95.45% (ideal farm power production is 16365.91 kW when wake calculation is not considered).

¹Each wind turbine is characterised by its cut-in (v_{cut-in}), rated (v_{rated}) and cut-out ($v_{cut-out}$) wind speed and its energy production is modelled accordingly. Reader may refer to Manwell et al. (2010); Marvuglia & Messineo (2012) for details.

²The data are freely available for research purposes and can be accessed from <http://www.irimo.ir/eng/index.php>

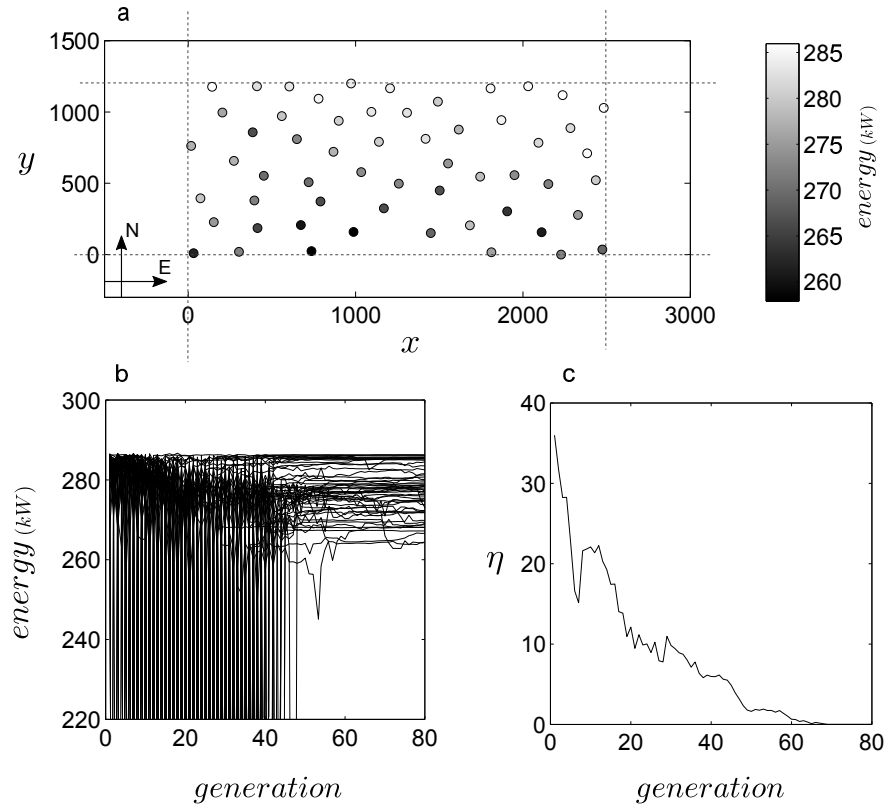


Figure 4: The position, power production of each turbine and convergence behaviour for Manjil wind farm design. The colorbar indicates the extent of each turbine's power production.

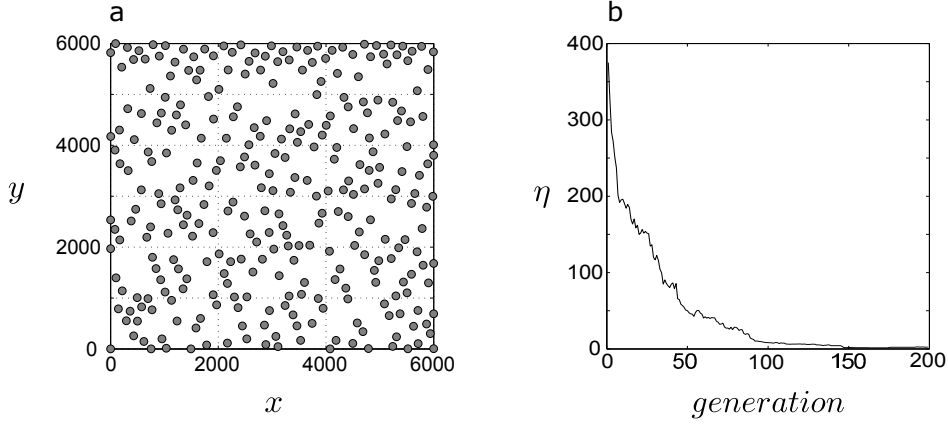


Figure 5: (a) The position, and (b) algorithm convergence behaviour for 300 wind turbines.

Table 2: The scalability performance analysis of the agent based approach for up to 300 wind turbines. The results are averaged over 30 runs.

n	Time (s)	Algorithm	kW - Power	kW - Ideal	efficiency
		generation	output (<i>st. dev</i>)	Power output	ζ
10	2.21	11	5184 (<i>0</i>)	5184	100.00%
40	28.16	115	19924.17 (<i>9.2</i>)	20736	96.08%
80	98.81	145	39391.40 (<i>39.5</i>)	41472	94.98%
150	354.22	122	72207.22 (<i>92.3</i>)	77760	92.86%
300	1402	148	139555.59 (<i>123.3</i>)	155520	89.73%

5. Further consideration

5.1. Scalability to high dimension problem - 300 wind turbines

To test the scalability of the proposed algorithm in higher dimension problems, we use the fixed intensity unidirectional wind of first test case problem and solve it for $n = 10, 40, 80, 150$, and $n = 300$ turbines (agents). Table 2 suggests a mild exponential trend in computational burden as the number of turbines (agents) increases. While for $n = 10$, it takes about 3 seconds on average to solve the problem (with average number of 11 generation before convergence), the average running time for $n = 300$ turbines reaches up to about 1400 seconds with 150 generation on average. Figure 5 shows the position of each turbine and the convergence behaviour of the algorithm in the largest case, where there are 300 turbines. As indicated in Table 2, turbine optimal positioning could achieve 90% efficiency with a promising convergence behaviour shown in Figure 5.

5.2. Centralised algorithm comparison

To compare the performance of a centralised aggregated optimisation problem, here called central planner (CP), with the proposed agent based approach, we use the above Manjil farm layout problem. Under CP approach, farm's total power production is a single aggregated problem defined by problem in equation 5. The search algorithm is differential evolution (DE) for both CP and agent based approaches with the identical wind, wake and power generation models introduced earlier. The population size for each turbine in agent based approach is kept at $m = 5$ and for CP approach the single population consists of 100 individuals. We run the algorithm for 200 generations, use η value as the performance metric and record the time taken for the whole simulation to study the convergence characteristics. Figure 6 shows the convergence behaviour. As shown, while agent based approach converged smoothly after only 70 generation in about 140 seconds, CP approach requires more than 150 generation with a substantially slower convergence time. The quick convergence of the agent based approach is achieved despite the higher number of individuals being evaluated ($57 \times 5 = 285$ individuals for agent based compared to 100 for CP approach). Further, the total wind power production under CP approach is 15291.87 kW which is about 4% less than 15962.51 kW power produced using the agent based method.

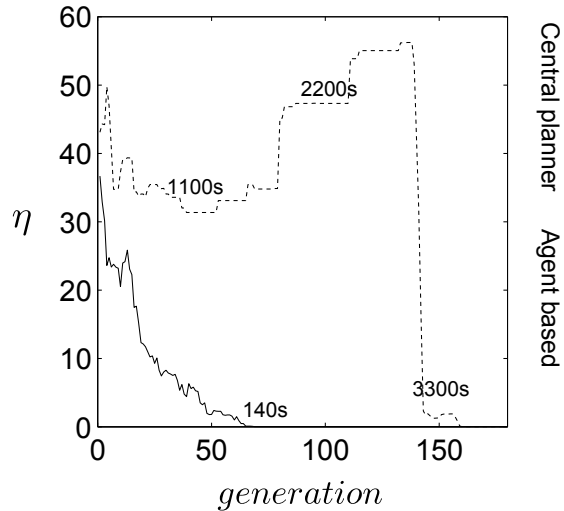


Figure 6: The convergence comparisons between centralised approach and decentralised agent based method. The values on each graph indicate the time taken in seconds to reach the corresponding generation.

6. Conclusion and future work

We modelled the wind farm design layout problem in a *bottom-up* decentralised way. This may be a valuable extension of currently used models for wind farm design as it can be easily scaled up for high dimension problems, and enables easier adaptation to complex wind farm environment. We represented

the turbines as self-adaptive agents that collectively solve a common task of maximising power production by communicating with each other while solving their own problem. In the proposed model, agents, here turbines, directly interact with their local neighbours and indirectly communicate with the others via the environment. The agents reward themselves by modifying their positions and by contributing to the total power production of the farm. We developed an evolutionary approach to solve these interrelated optimisation problems simultaneously. The quality of the results is satisfactory. The analysis on scalability of the algorithm up to 300 wind turbines shows a promising outcome. The performance of the algorithm when it is compared to a centralised algorithm indicates encouraging results in terms of convergence time and quality. In the proposed model, the numbers of wind turbines were fixed and the wind turbines were all identical. As a consequence, no economic or environmental aspects were considered and some parameters were disregarded such as air density, ground elevation and the height of the wind turbines. Considering these into the modelling as well as the sensitivity of the algorithm to its parameters are left for future work.

References

- Chehourri, A., Younes, R., Ilinca, A., & Perron, J. (2015). Review of performance optimization techniques applied to wind turbines. *Applied Energy*, 142, 361–388.
- Chowdhury, S., Zhang, J., Messac, A., & Castillo, L. (2012). Unrestricted wind farm layout optimization (uwflo): Investigating key factors influencing the maximum power generation. *Renewable Energy*, 38, 16–30.
- Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, 186, 311–338.
- Ding, N., Erfani, R., Mokhtar, H., & Erfani, T. (2016). Agent based modelling for water resource allocation in the transboundary Nile river. *Water*, 8, 139.
- Erfani, T., & Erfani, R. (2015a). An evolutionary approach to solve a system of multiple interrelated agent problems. *Applied Soft Computing*, 37, 40–47.
- Erfani, T., & Erfani, R. (2015b). Fair resource allocation using multi-population evolutionary algorithm. In *Applications of Evolutionary Computation* (pp. 214–224). Springer.
- Feng, J., & Shen, W. Z. (2015). Solving the wind farm layout optimization problem using random search algorithm. *Renewable Energy*, 78, 182–192.
- Forrest, S. (1990). Emergent computation: self-organizing, collective, and cooperative phenomena in natural and artificial computing networks: introduction to the proceedings of the ninth annual cnls conference. *Physica D: Nonlinear Phenomena*, 42, 1–11.
- González, J. S., Payán, M. B., Santos, J. M. R., & González-Longatt, F. (2014). A review and recent developments in the optimal wind-turbine micro-siting problem. *Renewable and Sustainable Energy Reviews*, 30, 133–144.
- Grady, S., Hussaini, M., & Abdullah, M. M. (2005). Placement of wind turbines using genetic algorithms. *Renewable energy*, 30, 259–270.
- Herbert-Acero, J. F., Probst, O., Réthoré, P.-E., Larsen, G. C., & Castillo-Villar, K. K. (2014). A review of methodological approaches for the design and optimization of wind farms. *Energies*, 7, 6930–7016.
- Jensen, N. O. (1983). *A note on wind generator interaction*.
- Katic, I., Højstrup, J., & Jensen, N. O. (1986). A simple model for cluster efficiency. In *European Wind Energy Association Conference and Exhibition* (pp. 407–410).

- Khan, S. A., & Rehman, S. (2013). Iterative non-deterministic algorithms in on-shore wind farm design: A brief survey. *Renewable and Sustainable Energy Reviews*, 19, 370–384.
- Kusiak, A., & Song, Z. (2010). Design of wind farm layout for maximum wind energy capture. *Renewable Energy*, 35, 685–694.
- Manwell, J. F., McGowan, J. G., & Rogers, A. L. (2010). *Wind energy explained: theory, design and application*. John Wiley & Sons.
- Marvuglia, A., & Messineo, A. (2012). Monitoring of wind farms power curves using machine learning techniques. *Applied Energy*, 98, 574–583.
- Mosetti, G., Poloni, C., & Diviacco, B. (1994). Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. *Journal of Wind Engineering and Industrial Aerodynamics*, 51, 105–116.
- Neri, F., & Tirronen, V. (2010). Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*, 33, 61–106.
- Pookpant, S., & Ongsakul, W. (2013). Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients. *Renewable Energy*, 55, 266–276.
- Price, K. V., Lampinen, J. A., & Storn, R. M. (2005). *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.
- Rahbari, O., Vafaeipour, M., Fazelpour, F., Feidt, M., & Rosen, M. A. (2014). Towards realistic designs of wind farm layouts: Application of a novel placement selector approach. *Energy Conversion and Management*, 81, 242–254.
- Sinha, A., Malo, P., Frantsev, A., & Deb, K. (2014). Finding optimal strategies in a multi-period multi-leader–follower stackelberg game using an evolutionary algorithm. *Computers & Operations Research*, 41, 374–385.
- Zhang, J., Chowdhury, S., Messac, A., & Castillo, L. (2013). A multivariate and multimodal wind distribution model. *Renewable Energy*, 51, 436–447.